View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Servicio de Difusión de la Creación Intelectual

Adaptive Two-Phase Spatial Association Rules Mining Method

Chin-Feng Lee and Mei-Hsiu Chen
Department of Information Management, Chaoyang University of Technology
168, Jifong E.Rd., Wufong Township, Taichung County 41349, Taiwan (R.O.C.)
Email : lcf@cyut.edu.tw

ABSTRACT

Since huge amounts of spatial data can be easily collected from various applications, ranging from remote sensing technology to geographical information system, the extraction and comprehension of spatial knowledge is a more and more important task. Many excellent studies on Remote Sensed Image (RSI) have been conducted for potential relationships of crop yield. However, most of them suffer from the performance problem because their techniques for mining association rules are based on Apriori algorithm. In this paper, two efficient algorithms, two-phase spatial association rules mining and adaptive two-phase spatial association rules mining, are proposed for address the above problem. Both methods primarily conduct two phase algorithms by creating Histogram Generators for fast generating coarse-grained spatial association rules, and further mining the fine-grained spatial association rules w.r.t the coarse-grained frequently patterns obtained in the first phase. Adaptive two-phase spatial association rules mining method conducts the idea of partition on an image for efficiently quantizing out non-frequent patterns and thus facilitate the following two phase process. Such two-phase approaches save much computations and will be shown by lots of experimental results in the paper.

Keywords: Spatial Database, Data Mining, Spatial Association Rules, Remote Sensed Image

1. INTRODUCTION

Those spatial data, from satellite photos or automated data gathering tool, such as remotely sensed images[21], local population, and what not have many widely-used sides of application, in many of today' research fields. When storing large amount of either spatial or non-spatial data in the spatial database, the past data gathering tool always emphasizes on how to improve on searching spatial data as well as storage system. However, there is very likely some knowledge or expertise potentially resided in this large amount of acquired data that is not discovered. With the help of spatial data mining, more interesting and useful information are discovered [3][7][8][9][10][11] [14][15][16]. This has, in turn, brought up a new and popular field for research today.

The methods in [4][5][6][18] for data mining on Remotely Sensed Images (RSI). Those methods are to take advantage of the association between reflectance intensities on RSI and crop yield through association rule. By looking at a profusion of remotely sensed images stored in geographical information system and considering the association between the RSI and crop yield at the same time, it applies data mining to excavate the useful expertise that may help the experts or agribusiness people in their difficulties to crop cultivation. But if applying data mining simply on discovering this knowledge from large amount of remotely sensed images, it does not sound very economical considering the time spent and efforts put in. So, designing an effective data mining method is very worthwhile for in-depth research.

More details on methods from the reference will be discussed in Section 2. Section 3 discusses how to acquire the association rule on RSI from two-phase association rules mining method. Section 4 presents a method for adaptive two-phase association rules mining. Section 5 verifies the validity of our proposed method through experimentation and discussion. Last section is our conclusion and future works.

2. ASSOCIATION RULES FOR SPATIAL DATABASES

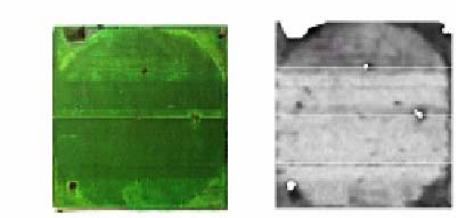
2.1 Association Rules

There are a number of researches going on the association rules mining [1][2][12][13][17] [19]. The association rule is "$X \rightarrow Y,(s,c)$" where X and Y are an itemset with X being the antecedent and Y being the consequent of rule. In example of the association between population and crop yield, S denotes the degree of support in which it represents the probability that X and Y occurs simultaneously, and c denotes the degree of confidence in which it represents the probability of Y occurrence when X occurs in $c = \frac{\text{support}(X \cup Y)}{\text{support}(X)}$. In some reference, support count $= s \times |D|$, where $|D|$ is the count in data table. Throughout our paper, the support count (*count*) will be taken as the count of frequent item sets. Decision makers can pre-define a minimum support count and minimum confidence in order for the elimination of infrequent item sets and the creation of association rules through frequent item sets.

2.2 Spatial Data Mining on RSI

Images on RSI can be parted into a couple of different types such as TM, SPOT, AVHRR, and TIFF. Take TM image on RSI as example, TM consists of seven bands which are B for Blue, G for Green, R for Red, RIR for Reflective-Infrared, MIR for Mid-Infrared, TIR for Thermal-Infrared, and MIR2 for Mid-Infrared2[4][5][6][18]. Each band contains a relative reflectance intensity value in the range 0-to 255 for each pixel. Data mining application in the past always bases its viewpoint on the agricultural yield. One remotely sensed image is associated with yield, in the way which that one yield image is a map that defines agricultural yield standards. It is often shown in color or gray-scaled image. Shown in Figure 1 (a) is a remotely sensed image and (b) is a yield image.

Incorporating the association between remotely sensed image and yield, data mining is to excavate the useful expertise that may help the experts or agribusiness people to improve crop cultivation.



(a) (b)

Figure 1: Remotely sensed image and Yield map

3. TWO-PHASE SPATIAL ASSOCIATION RULES MINING METHOD

Figure 2 illustrates our proposed two-phase association rule architecture for RSI data mining. First, acquire an image from an RSI database. Do a color count and analysis of variance on this image and generate the Histogram Generator (HG) from the user-predefined intervals. HG looks for the most representative characteristic value in an image to quickly find the coarse-grained association rules. From these coarse-grained association rules, fine-grained association rules can be obtained for the RSI database. This method is primarily divided into four steps. First step is color count and analysis of variance on remotely sensed images. Second step is to generate Histogram Generator according to user-predefined intervals. Third step is to mine out the coarse-grained association rules according the user generated HG through algorithm of association rules. Fourth step is to mine out the fine-grained association rules from the spatial database according to the coarse-grained association rules. The procedure on two-phase association rule for RSI mining as follows:

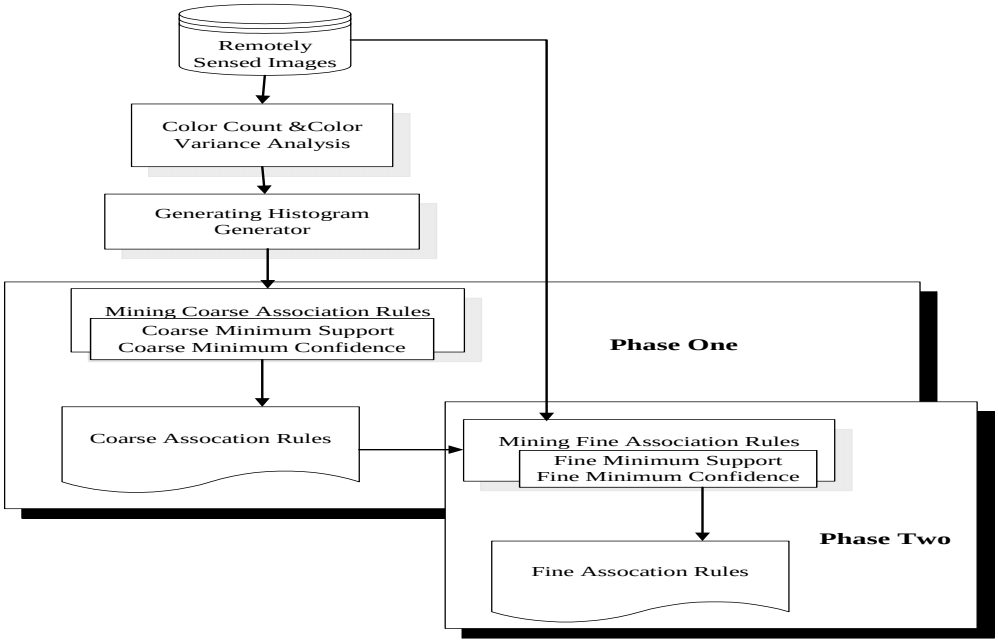


Figure 2. Two-phase association rule architecture for RSI data mining

Step 1: Color Count and Analysis of Variance

In a RSI spatial database D , there are $\{P_1, P_2, \ldots, P_r\}$ remotely sensed images in which every RSI P_i consists of $n \times n$ pixels. First, calculate the pixel color count from each P_i; and of each P_i, T_i color count is produced.

Following is the analysis of variance. Pick a P_i and divide into m number of small blocks. Calculate its pixel color count as $T_{i1}, T_{i2}, \ldots, T_{im}$ from each small block $P_{i1}, P_{i2}, \ldots, P_{im}$. Take the average from pixel color count of *m* block. The formula is shown as follows:

$$\overline{T_i} = \sum_{j=1}^{m} T_{ij} / m , \qquad (1)$$

Where T_{ij} denotes the pixel color count of m number of small blocks of P_i and $\overline{T_i}$ the average pixel color count of P_i after division.

Then calculate the degree of variance from color of this remotely sensed image; the variance formula is as follows:

$$V_i = \sum_{j=1}^{m} (T_{ij} - \overline{T_i}) / m, \qquad (2)$$

Where T_{ij} denotes the pixel color count of m number of small blocks of P_i, $\overline{T_i}$ the average pixel color count of P_i after division, and V_i the color variance of P_i.

Example 3-1： Take Figure 3 for example calculate its pixel color count of this RSI. Next this image is divided into nine small blocks. Now calculate the color count of

each block assuming that the color counts for these blocks from 1 to 9 in the RSI are 20, 3, 10, 8, 7, 20, 2, 10 and 1, respectively. On to the formula (1), the average pixel color count of the RSI after division is calculated in $(20+3+10+8+7+20+2+10+1)/9=9$. Hence, the color variance of this RSI based on formula (2) is calculated in $[(20-9)^2+(3-9)^2+\ldots+(1-9)^2]/9=44.22$.

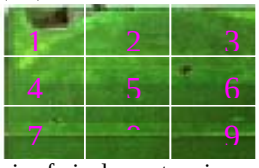


Figure 3. Analysis of pixel count variance in RSI

Step 2: Generate Histogram Generator

The main purpose of Histogram Generator (HG) reassigns the intensity of pixel for each pixel in the image. It means using the small intensity of pixel to substitute for the large intensity of pixel. For example, assume we want to transform the 24-bit high-color image into 256(2^8) gray image. First, we create HGs of 256 gray levels. Then, an input image with 16,770,000(2^{24}) high-color levels may be quantized to an image with only 256 gray levels.

The main purpose of Histogram Generator is to look for the most representative characteristic value in images. For a RGB image in a 3-dimensional color space, each color value in every dimension ranges from 0-to-255. In this step, each color value in every dimension is divided into k intervals $S_0, S_1, \ldots, S_{k-1}$ in which they are $S_0=\left[0, \left\lceil \frac{255}{k} \right\rceil\right]$, $S_1=\left[\left(\left\lceil \frac{255}{k} \right\rceil+1\right), 2\left\lceil \frac{255}{k} \right\rceil\right]$, …, $S_{k-1}=\left[((k-1)\left\lceil \frac{255}{k} \right\rceil+1), 255\right]$, respectively. Because of that HG defines each color value in every interval, there are, in every dimension, k HGs such as $HG_0, HG_1, \ldots, HG_{k-1}$. Take the medial value in each interval to be its representative value, the medial values as $M_0, M_2, \ldots, M_{k-1}$ are thus produced.

Example 3-2: Take k =4 as instance, it defines HG_0, HG_1, HG_2 and HG_3 as [0, 64], [65, 128], [129, 192] and [193, 255], respectively. Its median values are 32, 96.5, 160.5 and 223.5, respectively as shown in Table 1.

Table 1. Four HGs are produced for a certain color dimension

Histogram Generator	HG_0	HG_1	HG_2	HG_3
Interval (S_i)	[0, 64]	[65, 128]	[129, 192]	[193, 255]
Median Value(M_i)	32	96.5	160.5	223.5

Step 3: Discovery of Coarse-grained Association Rules according to Histogram Generator

Assume there are two images, remotely sensed image and yield image, each of which is of $n \times n$ pixels. The corresponding data table T^{RSI} for the RSI contains $coordinate, band_1, band_2, \ldots, band_j$, where *coordinate* is the position of pixels and $band_1, band_2, \ldots band_j$ has different definitions depending on what types in the RSI shows. From the type of TM on RSI shows, it consists of seven bands which are B, G, R, RIR, MIR, TIR and MIR2. For each tuple value t_j of $band_j$ are in the ranges from 0-to-255 ($t_j \in [0, 255]$), it is specified $band_j^{t_j}$. Equally the corresponding data table T^{Yield} for the yield image contains *coordinate*,*Yield*, where *coordinate* is the position of pixels, *Yield* is the yield standard, and each tuple value t' of *Yield* are in the range from 0-to-255, it is specified $Yield^{t'}$. For example, R^{255} is the value 255 of band R. Based on their positions of coordinates, the RSI and yield image are incorporated into one data table, namely T^{origin}, it consists of *Id*, *coordinate*, $band_1$, $band_2$, …, $band_j$, and *Yield*, where its *Id* is the item number of the table as shown in Table 2. From the generated HGs by each pixel *i*, find every representative value of HG_k for replacement, to which each value of $band_j$ and Yield, known as $band_j^{t_j}$ and $Yield^{t'}$, belongs to. The concept here lies is find the minimum variance in the comparison between the value of $band_j^{t_j}$ and M_k of HG_k, from which the k in this HG replaces $band_j^{t_j}$. Formula for its calculation is as follows:

$$d(HG_k, band_j^{t_j}) = \left|\Delta(M_k - band_j^{t_j})\right|, \quad (3)$$

$$d(HG_k, Yield^{t'}) = \left|\Delta(M_k - Yield^{t'})\right|, \quad (4)$$

Where HG_k represents the k-th HG, $band_j^{t_j}$ is the value of $band_j$ in one pixel *i*, $Yield^{t'}$ is the value of Yield in one pixel *i* and M_k the k-th representative value of HG.

The table T^{HG} is produced after each pixel in an image goes through the above HG quantized. The table T^{HG} consists of *Id*, *coordinate*, $band_1$, $band_2$, …, $band_j$, and *Yield*, as show in Table 3.

Example 3-3： Suppose there are one 3×3 remotely sensed image and another 3×3 yield image; and in the RSI there are 3 bands – R, G, and B that are combined with Yield in the yield image as shown in Table 2. The user-defined intervals generate 4 Histogram Generators as shown in Table 1.

Table 2. Original 3×3 image data table

Id	coordinate	R	G	B	Yield
1	0, 0	255	255	255	255
2	0, 1	255	255	255	255
3	0, 2	255	255	255	255
4	1, 0	255	200	50	200
5	1, 1	200	255	255	200
6	1, 2	200	255	255	200
7	2, 0	100	200	255	200
8	2, 1	100	200	255	255
9	2, 2	100	200	255	255

In example of coordinate (2, 0), R=100 from the formula (3) and (4). Other results after calculations are |32-100|=68 from HG_0, |96.5-100|=3.5 from HG_1, |160.5-100|=60.5 from HG_2, and |223.5-100|=123.5 from HG_3. Hence, R=100 belongs to HG_1.

Table 3. Quantized 3×3 image data table

Id	*coordinate*	*R*	*G*	*B*	*Yield*
1	0, 0	3	3	3	3
2	0, 1	3	3	3	3
3	0, 2	3	3	3	3
4	1, 0	3	3	0	3

5	1, 1	3	3	3	3
6	1, 2	3	3	3	3
7	2, 0	1	3	3	3
8	2, 1	1	3	3	3
9	2, 2	1	3	3	3

Then, start data mining on the T^{HG}. The process of data mining is based on Apriori's algorithm [1]. And in mining research of association rules on RSI, the relationship between the RSI and yield is of our main concern in order for better understanding the impact of reflectance intensities on RSI brings on the yield. The relationship of these association rules is antecedent-and-consequent. As mentioned above, we have to modify Apriori. As the Algorithm 1 indicates, the reflectance intensities on RSI are the antecedent, and the yield is the consequent part which is then added to the resulting itemset candidate. This itemset must have its count larger than the minimum support(s^c) to be the frequent itemset. Each item in frequent 1-itemsets (F_1) is represented as $I(band_j^{S_\alpha}, Yield^{S_\alpha'})$, because the itemsets in F_1 can be represented as an association rule that consists of a antecedent and a consequent. For the association rule by the itmsets I, its antecedent is represented as $I.band_j^{S_\alpha}$ and consequent as $I.Yield^{S_\alpha'}$. Further calculate frequent ℓ-itemsets(F_ℓ), assume $\theta_1 \in F_{\ell-1}$ and $\theta_2 \in F_{\ell-1}$ where $\theta_1 \neq \theta_2$. To produce frequent ℓ-itemsets (F_ℓ), first step has to consider whether or not $\theta_1.Yield^{S_\alpha'}$ equals to $\theta_2.Yield^{S_\alpha'}$. And if they equal to each other, they are joined together to produce candidate ℓ-itemsets(C_ℓ). In candidate ℓ-itemsets (C_ℓ), each format of the itemsets is $I(band_1^{S_{\alpha_1}}, band_2^{S_{\alpha_2}}, \ldots, band_{\ell-2}^{S_{\alpha_{\ell-2}}}, band_{\ell-1}^{S_\beta}, band_\ell^{S_\gamma}, Yield^{S_\alpha'})$; of which $I(band_1^{S_{\alpha_1}}, band_2^{S_{\alpha_2}}, \ldots, band_{\ell-2}^{S_{\alpha_{\ell-2}}}, band_{\ell-1}^{S_\beta}, band_\ell^{S_\gamma})$ is the antecedent, $I.Yield^{S_\alpha'}$ is the consequent. Similarly, frequent ℓ-itemsets (F_ℓ) will be produced. frequent ℓ-itemsets are formatted as $I(band_1^{S_\alpha}, band_2^{S_\alpha}, \ldots, band_\ell^{S_\alpha}, Yield^{S_\alpha'})$, $I.band_1^{S_\alpha}, I.band_2^{S_\alpha}, \ldots, I.band_\ell^{S_\alpha}$ is represented as the antecedent, $I.Yield^{S_\alpha'}$ as the consequent. For example, the frequent 3-itemsets of $I(R^0, G^0, B^0, Yield^1)$, the intensity of pixel of R is 0, of G is 0, of B is 0, of Yield is 1. When the support count is larger than the minimum support count, The resulting association rules at last are as follows:

$$band_1^{S_\alpha}, band_2^{S_\alpha}, \ldots, band_j^{S_\alpha} \Rightarrow Yield^{S_\alpha'}, count = s_{\Re}, CF = CF_{\Re}, ID_{\Re},$$

Where $band_1^{S_\alpha}, band_2^{S_\alpha}, \ldots, band_j^{S_\alpha}$ represents as the antecedent part of frequent itemset mined by T^{HG}, $Yield^{S_\alpha'}$ as the consequent part of frequent item sets mined by T^{HG} which is the Yield, *count* as the count of frequent itemset, *CF* as the confidence, and $ID_{\Re}$ as the record of *Id* position of frequent itemset in T^{origin}, $ID_{\Re} = \{Id | Id \in 1, 2, \ldots, n \times n\}$. □

Algorithm 1. Find frequent coarse-grained_itemsets

Input: (1) T^{HG}: A quantized table in which each value of bands and Yield is transformed by a set of histogram generators $\{HG_0, HG_1, \ldots, HG_{k-1}\}$
/* each tuple in T^{HG} is formated as (*Id, coordinate, band*$_1$, *band*$_2$, …, *band*$_j$, *Yield*) */.
(2) s^c: Coarse-grained minimum support threshold.
Output: F: a set of frequent itemsets in T^{HG}

Step1:
For each $band_j$ and *Yield*
If the tuple counts for $band_j^{S_\alpha}$ and $Yield^{S_\alpha'}$ satisfy s^c.
then put $(band_j^{S_\alpha}, Yield^{S_\alpha'})$ into F_1; /* F_1 is a 1-frequnet itemset with pairs of one $band_j$ and *Yield* */

Step2:
For (ℓ =2 ; $F_{\ell-1} \neq \varnothing$; ℓ ++)
C_ℓ =**apriori_antecedent_consequent**($F_{\ell-1}$)
For each tuple $t \in T^{HG}$
C_t=subset(C_ℓ, t)
For each candidate $c \in C_t$ /* Read each tuple and count the c.count++ occurrences of each item*/
$F_\ell = \{c \in C_t \,|\, c.\text{count} \geq s^c\}$
return $F = \cup_\ell F_\ell$

End of Algorithm 1.

Procedure aprioir_antecedent_consequent($F_{\ell-1}$ **) /* join_step*/**
For each itemset $\theta_1 \in F_{\ell-1}$
For each itemset $\theta_2 \in F_{\ell-1}$
If $((\theta_1.Yield^{S_\alpha'}) = (\theta_2.Yield^{S_\alpha'}))$
If$((\theta_1.band_1^{S_{\alpha_1}}) = (\theta_2.band_1^{S_{\alpha_1}})) \cap ((\theta_1.band_2^{S_{\alpha_2}}) = (\theta_2.band_2^{S_{\alpha_2}})) \cap \ldots \cap ((\theta_1.band_{\ell-2}^{S_{\alpha_{\ell-2}}}) = (\theta_2.band_{\ell-2}^{S_{\alpha_{\ell-2}}})) \cap ((\theta_1.band_{\ell-1}^{S_\beta}) < (\theta_2.band_\ell^{S_\gamma}))$
then $\{c = (band_1^{S_{\alpha_1}}, band_2^{S_{\alpha_2}}, \ldots, band_{\ell-2}^{S_{\alpha_{\ell-2}}}, band_{\ell-1}^{S_\beta}, band_\ell^{S_\gamma}, Yield^{S_\alpha'})$/* c is a candidate ℓ -itemset */
If has_infrequent_subset(c, $F_{\ell-1}$) then delete c;
else add c to C_ℓ;}
return C_ℓ;

Procedure has_infrequent_subset(*c*, $F_{\ell-1}$**)**
For each (ℓ *-1*)-subset δ of c
If $\delta \notin F_{\ell-1}$ then return TRUE;
return FALSE;

Example 3-4: Again take **Example 3-3** as an example; the entire original image after these calculations is represented in Table 3. Next to obtain coarse-grained spatial association rules as shown in Figure 4, apply the mining on the association rules from the results of calculations.

- $R^{[193, 255]} \Rightarrow Yield^{[193, 255]}$, count=6, CF=1, ID=[1, 2, 3, 4, 5, 6]
- $R^{[65, 128]} \Rightarrow Yield=^{[193, 255]}$, count=3, CF=1, ID=[7, 8, 9]

• • • •

- $R^{[193, 255]}$, $B^{[193, 255]} \Rightarrow Yield^{[193,255]}$, count=5, CF=1, ID=[1, 2, 3, 5, 6]
- $R^{[65,128]}$, $G^{[193,255]}$, $B^{[193, 255]}$ $\Rightarrow Yield^{[193,255]}$,count=3,CF=1,ID=[7, 8, 9]

Figure 4. Coarse-grained association rules

Step 4: Mining for fine-grained association rules

The main purpose of this step – Step 4 is to obtain the fine-grained association rules. From the mined coarse-grained association rules $R_1, R_2, \ldots, R_{\Re}$ in Step 3, the following is obtained: $ID_1, ID_2, \ldots, ID_{\Re}$. Each $ID_{\Re}$ then contains $Id_1^1, Id_2^2, \ldots Id_{\Re}^d$ which record the *Id* positions of these association rules in original data table T^{origin}. Go back to the original data table T^{origin}, and calculate the count of value $band_j^{t_j}$ in each band; delete every calculated count that is smaller than minimum support count $band_j^{t_j}$, a fine-grained association rule is now obtainable. The processes are presented in Algorithm 2.

Algorithm 2. Find frequent fine-grained_itemsets

Input: $\Re$: a coarse-grained association rule

s^f: fine-grained minimum support threshold.

Output: F^f: frequent fine-grained-itemsets

Let $\Re$ be $band_1^{S_\alpha}$, $band_2^{S_\alpha}$, …, $band_j^{S_\alpha} \Rightarrow Yield^{S_\alpha}$', $count = s_{\Re}$, $CF = CF_{\Re}$, $ID_{\Re}$,

Where $ID_{\Re} = \{Id | Id \in 1, 2, \ldots, n \times n\}$; $\alpha \in \{0, 2, \ldots, k-1\}$, k is the number of Histogram Generator

Step 1: Let T^{temp} be a temp table in which each tuple comes from those in T^{origin} w.r.t. $ID_{\Re}$. Each tuple in T^{temp} is formated as (*Id, coordinate, band*$_1$*, band*$_2$*, …, band*$_j$*, Yield*.

Step 2:

Insert into $F_j(band_1, band_2, \ldots, band_j, count)$

Select $T^{temp}.band_1, T^{temp}.band_2, \ldots, T^{temp}.band_j, count(*)$

From T^{temp}

Group by $T^{temp}.band_1, T^{temp}.band_2, \ldots, T^{temp}.band_j$

Having $count(*) >= s^f$

End of Algorithm 2.

Example 3-5: Again take **Example 3-4** as an example; for instance on the first association rule $R^{[193, 255]} \Rightarrow Yield^{[193, 255]}$, count=6, CF=1, ID=[1, 2, 3, 4, 5, 6] in Figure 4, its ID is [1, 2, 3, 4, 5, 6]. Back to the original table – Table 2, do a separate calculation on the original data in Band R and Yield, providing that either of which Id is only 1, 2, 3, 4, 5, 6. Whichever case satisfies the minimum support count brings in the fine-grained association rule $R^{255} \Rightarrow Yield^{255}$. Another instance on the last association rule $R^{[65, 128]}$, $G^{[193, 255]}$, $B^{[193, 255]} \Rightarrow Yield^{[193,255]}$, count=3, CF=1, ID=[7, 8, 9], similar calculation as in the previous instance goes for the ID being [7, 8, 9]; in this case, the fine-grained association rule R^{100}, G^{200}, $B^{255} \Rightarrow Yield^{255}$ is obtained.

4. ADAPTIVE TWO-PHASE SPATIAL ASSOCIATION RULES MINING METHOD

The two-phase data mining of spatial association rules in section 3 can improve the Apriori method in terms of its efficiency. However during the image processing, partitioning the image into parts can improve its efficiency; to this point, the concept of image blocking is incorporated onto the coarse-grained two-phase data mining of spatial association rules. Image blocking performs the mining on each of the disjoining blocks partitioned from an image. The motive of taking on the image blocking is due to the inter-pixel redundancy in image data [20]. That means, the occurrences of neighboring image points redundancy is quite high; in other words, there is a higher possibility of more frequent itemsets. Therefore, the adaptive two-phase data mining of spatial association rules is to eliminate the blocks which do not produce frequent itemsets to improve its efficiency through image blocking.

4-1. Flowchart of the adaptive two-phase data mining of spatial association rules

Our enhanced method has been made from some improvements over the method by Savasere et al. [19]. Figure 5 is the flowchart of coarse-grained association rules through image blocking. Exploration of the frequent itemsets in coarse-grained association rules is made in four main steps. First step is to partition an image. Second step is to produce local frequent itemsets and record local non-frequent itemsets at the same time. Third step is to group each local frequent itemsets into global candidate itemsets. Fourth step is to add together the support count in global candidate itemsets as well as in non-frequent itemsets to be the final support count in global candidate itemsets. When bigger than the minimum support count, a global frequent itemsets is produced. Next sub-chapter will put more emphasis on each of the four steps.

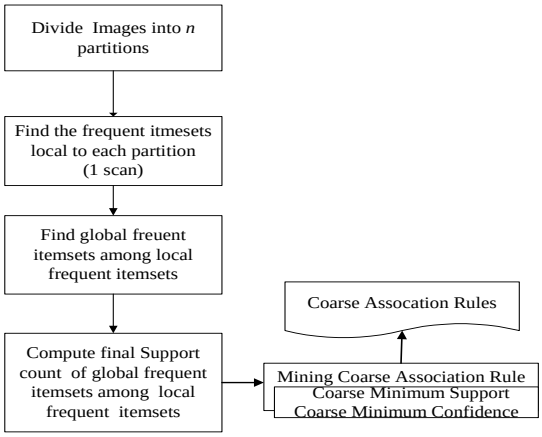


Figure 5. Adaptive two-phase association rule flowchart for RSI data mining

4-2. Procedures of the enhanced two-phase data mining of spatial association rules

Step 1: Partition the image

According to the mined coarse-grained association rules by Histogram Generator in Step 3 of Chapter 3, T^{HG} is the data table through HGs quantization of T^{origin}. Perform data mining on T^{HG} to explore the

coarse-grained association rules. And the entire mining process is shown in Algorithm 3. First is to partition into n number of disjoining blocks $\{P^1, P^2, \ldots, P^n\}$. Divide the minimum support count by the count in data table T^{HG}. Multiply what's left in the previous calculation by the count in P^i to obtain the minimum support count in each block ($s_p^c = \frac{s^c}{|T^{HG}|} \times |p^i|$), where s_p^c is the minimum support count in the block, $|T^{HG}|$ is the count of data table T^{HG}, s^c is the minimum support count and $|p^i|$ is the count in the block.

Example 4.1: Table 4 is an image of 8×2 pixels, after which the data table T^{HG} is made from quantization by four HGs on bands R, G, B and Yield, respectively. The image is partitioned into four small blocks (n=4) in which block P^1 consists of {1, 2, 3, 4}, block P^2 of {5, 6, 7, 8}, block P^3 of {9, 10, 11, 12}, and block P^4 of {13, 14, 15, 16}. Whereas minimum support count (S^c) is 10, the minimum support count in the block is $s_p^c = \frac{s^c}{|T^{HG}|} \times |p^1| = \frac{s^c}{|T^{HG}|} \times |p^2| = \frac{s^c}{|T^{HG}|} \times |p^3| = \frac{s^c}{|T^{HG}|} \times |p^4| = \frac{10}{16} \times 4 = 2.5$.

Table 4. Four partitioned blocks 8×2 image data table

Id	coordinate	R	G	B	Yield
1	0,0	3	3	3	3
2	0,1	0	0	0	3
3	1,0	3	3	3	3
4	1,1	3	3	3	3
5	2,0	3	3	3	3
6	2,1	0	0	0	3
7	3,0	0	0	0	3
8	3,1	1	1	1	3
9	4,0	3	3	3	3
10	4,1	3	3	3	3
11	5,0	3	3	3	3
12	5,1	3	3	3	3
13	6,0	3	3	3	3
14	6,1	3	3	3	3
15	7,0	3	3	3	3
16	7,1	3	3	3	3

Step 2: Exploration of local frequent itemsets in each block

Step2.1: Calculate local frequent 1-itemsets (F_1^i) for each P^i. And each item in local frequent 1-itemsets is represented as $I(band_j^{S_\alpha}, Yield^{S_\alpha'}, count)$ because the itemsets in F_1^i can be represented as an association rule that consists of a antecedent and a consequent. For the association rule by the itemsets I, its antecedent is represented as $I.band_j^{S_\alpha}$ and consequent as $I.Yield^{S_\alpha'}$. *I.count* is the support count of the itemset. For example, $I(R^0, Yield^1, 5)$ represents the color value of R as 0, the color value of Yield as 1 and the support count equals to 5.

On the other hand, record all the local non-frequent 1-itemsets that are smaller than the minimum support count in the block onto NF_1^i. Similarly, the format of local non-frequent 1-itemsets is $I(band_j^{S_\alpha}, Yield^{S_\alpha'}, count)$, where $I.band_j^{S_\alpha}$ is represented as the antecedent, $I.Yield^{S_\alpha'}$ as the consequent and *I.count* as the support count of the itemsets.

Step2.2: Assume $\theta_1 \in F_{\ell-1}^i$ and $\theta_2 \in F_{\ell-1}^i$ where $\theta_1 \neq \theta_2$. To produce local frequent ℓ-itemsets (F_ℓ^i), first step has to consider whether or not $\theta_1.Yield^{S_\alpha'}$ equals to $\theta_2.Yield^{S_\alpha'}$. And if they equal to each other, they are joined together to produce local candidate ℓ-itemsets (C_ℓ^i). In local candidate ℓ-itemsets, each format of the itemsets is $I(band_1^{S_{\alpha_1}}, band_2^{S_{\alpha_2}}, \ldots, band_{\ell-2}^{S_{\alpha_{\ell-2}}}, band_{\ell-1}^{S_\beta}, band_\ell^{S_\gamma}, Yield^{S_\alpha'}, count)$; of which $I(band_1^{S_{\alpha_1}}, band_2^{S_{\alpha_2}}, \ldots, band_{\ell-2}^{S_{\alpha_{\ell-2}}}, band_{\ell-1}^{S_\beta}, band_\ell^{S_\gamma}, Yield^{S_\alpha'}, count)$ is the antecedent, $I.Yield^{S_\alpha'}$ is the consequent and *count* is the support count=min($\theta_1.count$, $\theta_2.count$) of these established association rules in the local candidate itemsets. When the support count is larger than the minimum support count in the block, local frequent ℓ-itemsets (F_ℓ^i) will be produced. local frequent ℓ-itemsets are formatted as $I(band_1^{S_\alpha}, band_2^{S_\alpha}, \ldots, band_\ell^{S_\alpha}, Yield^{S_\alpha'}, count)$. Similarly, $I.band_1^{S_\alpha}, I.band_2^{S_\alpha}, \ldots, I.band_\ell^{S_\alpha}$ is represented as the antecedent, $I.Yield^{S_\alpha'}$ as the consequent and *I.count* as the support count of the itemsets. For example, $\theta_1 = (R^0, G^0, Yield^1, 4) \in F_2^i$, $\theta_2 = (R^0, B^0, Yield^1, 3) \in F_2^i$. When $\theta_1.Yield^1 = \theta_2.Yield^1$, $I(R^0, G^0, B^0, Yield^1, 3)$ will be produced and support count=min(4, 3); again when the support count is larger than the minimum support count in the block, the global frequent itemsets of $I(R^0, G^0, B^0, Yield^1, 3)$ will be produced. The color value of R is 0, of G is 0, of B is 0, of Yield is 1, and the support count is equal to 3.

On the other hand, record all the local non-frequent ℓ-itemsets that are smaller than the minimum support count in the block onto NF_1^i. Similarly, the format of local non-frequent ℓ-itemsets is $I(band_1^{S_\alpha}, band_2^{S_\alpha}, \ldots, band_\ell^{S_\alpha}, Yield^{S_\alpha'}, count)$, where $I.(band_1^{S_\alpha}, band_2^{S_\alpha}, \ldots, band_\ell^{S_\alpha})$ is represented as the antecedent, $I.Yield^{S_\alpha'}$ as the consequent and *I.count* as the support count of the itemsets.

Example 4.2: Take Table 9 for example, the block P^3 in which local frequent 1-itemsets are produced, (R^3, $Yield^3$, 4), (G^3, $Yield^3$, 4) and (B^3, $Yield^3$, 4) are also produced. Furthermore, the local frequent 2-itemsets are (R^3, G^3, $Yield^3$, 4) where count=4 obtained from min(4, 4).

On the other hand, no local frequent itemsets are produced in the block P^2 because the support count of the itemsets is smaller than the minimum support count in the block. Our method records the itemsets onto the local non-frequent itemsets so as to produce the local non-frequent itemsets such as (R^3, $Yield^3$, 1), (G^3. $Yield^3$, 1), and (R^3, G^3, $Yield^3$, 1).

Step 3: Composition of a global candidate itemsets from each local frequent itemsets

First step is to incorporate the local frequent ℓ-itemsets $F_\ell^1, F_\ell^2, \ldots, F_\ell^n$ in each block into a global candidate ℓ-itemsets (C_ℓ). Second is to calculate the support count in C_ℓ ; that is, $C_\ell.count = F_\ell^1.count + F_\ell^2.count + \ldots + F_\ell^n.count$.

Example 4.3: In Table 9, the local frequent 3-itemsets in P^1 are (R^3, G^3, B^3, $Yield^3$, 3), and the local frequent 3-itemsets in P^3 and P^4 are (R^3, G^3, B^3, $Yield^3$, 4). Incorporation of both comes the global candidate 3-itemsets which are (R^3, G^3, B^3, $Yield^3$, 11), where count=(3+4+4)=11.

Step 4: Calculation of the final support counts in global candidate itemsets for establishing the global frequent itemsets

Adding together the established global candidate itemsets $C_1, C_2, \ldots, C_\ell$ in Step 3 and the support count in each local non-frequent itemsets (NF_ℓ^i) yields $C_\ell.count = C_\ell.count + NF_\ell^i.count$, and the final support count in the global candidate itemsets can be produced. If the support count is larger than the minimum support count (s^c), then they are the global frequent itemsets.

Example 4.4: Again take Table 9 as an example, the global candidate itemsets (R^3, G^3, B^3, $Yield^3$, 11) are produced in Step 3. There are local non-frequent itemsets such as (R^3, G^3, B^3, $Yield^3$, 1) in P^2. Thus the final global candidate itemsets that are produced are (R^3, G^3, B^3, $Yield^3$, 12), where count=(11+1)=12 that is larger than the minimum support count 10. So the global frequent itemsets (R^3, G^3, B^3, $Yield^3$, 12) are produced.

Algorithm 3. Image partition for find frequent coarse-grained itemsets

Input: (1) T^{HG}: A quantized table in which each value of bands and Yield is transformed by a set of histogram generators {HG_0, HG_1, …, HG_{k-1}}
/* each tuple in T^{HG} is formated as (*Id, coordinate, band*$_1$, *band*$_2$, …, *band*$_j$, *Yield*) */.

(2) s^c: Coarse-grained minimum support threshold.

(3) η : Number of disjoin partitions

Output: F: a set of frequent itemsets in T^{HG}

Step 1:

Let P^i is a partitioned table from T^{HG}, $s_p^c = \frac{s^c}{|T^{HG}|} \times |p^i|$ is coarse-grained minimum support threshold for each partition table.

For each partition P^i */* for i=1, 2, …, n*/*

F_ℓ^i =gen_large_itemsets(P^i , s_p^c)

Step 2: /* Merge frequent itemsets of each partition and Compute counts of frequent-itemsets*/

For (ℓ =1; $F_\ell^i \neq \varnothing$, i=1, 2, …, n; ℓ ++)

$C_\ell = \cup_{i=1, 2, \ldots, n} F_\ell^i$ //Merge phase

$C_\ell.count = F_\ell^1.count + F_\ell^2.count + \ldots + F_\ell^n.count$

Step 3: /* Compute final counts of frequent-itemsets */

For all candidates C_ℓ

For each NF_ℓ^i in P^i

$C_\ell.count = C_\ell.count + NF_\ell^i.count$

End-For

$F = \{c \in C_\ell \mid c.count >= s^c\}$

End-For

return $F = \cup_\ell F_\ell$

End of Algorithm 3.

Procedure gen_large_itemsets(P^i , s_p^c)

Step1:

For each $band_j$ and *Yield*

If the tuple counts for

$band_j^{S_\alpha}$ and $Yield^{S_{\alpha'}}$ satisfy s_p^c.

then put ($band_j^{S_\alpha}$, $Yield^{S_{\alpha'}}$, *count*) into F_1^i

else

put ($band_j^{S_\alpha}$, $Yield^{S_{\alpha'}}$, *count*) into NF_1^i ;

End-For

Step 2: /* join_step*/

For (ℓ =2 ; $F_{\ell-1}^i \neq \varnothing$; ℓ ++)

For each itemset $\theta_1 \in F_{\ell-1}^i$

For each itemset $\theta_2 \in F_{\ell-1}^i$

If $((\theta_1.Yield^{S_{\alpha'}}) = (\theta_2.Yield^{S_{\alpha'}}))$

If

$((\theta_1.band_1^{S_{\alpha_1}}) = (\theta_2.band_1^{S_{\alpha_1}})) \cap ((\theta_1.band_2^{S_{\alpha_2}})$

$= (\theta_2.band_2^{S_{\alpha_2}})) \cap \ldots \cap ((\theta_1.band_{\ell-2}^{S_{\alpha_{\ell-2}}}$

$= (\theta_2.band_{\ell-2}^{S_{\alpha_{\ell-2}}})) \cap ((\theta_1.band_{\ell-1}^{S_\beta}) < (\theta_2.$

$band_\ell^{S_\gamma}))$ then { c^i = ($band_1^{S_{\alpha_1}}$, $band_2^{S_{\alpha_2}}$, …, $band_{\ell-2}^{S_{\alpha_{\ell-2}}}$, $band_{\ell-1}^{S_\beta}$, $band_\ell^{S_\gamma}$, $Yield^{S_{\alpha'}}$)

/* c^i is a candidate ℓ-itemset in P^i */

If has_infrequent_subset(c^i, $F^i_{\ell-1}$) then delete c^i;

else $c^i.count=min(\theta_1.count,\theta_2.count)$

If $c^i.count >= s^c_p$ then $F^i_\ell = F^i_\ell \cup \{c^i\}$

else $NF^i_\ell = NF^i_\ell \cup \{c^i\}$

return F^i_ℓ ;

Procedure has_infrequent_subset(c^i; $F^i_{\ell-1}$)

/* prune_step*/

For each (ℓ -1)-subset δ of c^i

If $\delta \notin F^i_{\ell-1}$ then

return TRUE;

return FALSE;

5. EXPERIMENT AND DISCUSSION

5-1. Experimental results from the two-phase data mining of spatial association rules

To conduct the experiment, there are *k* numbers of different images prepared. Each of these images is 50,000 pixels. And with the various kinds of Histogram Generators, an observation can be made on the increased efficiency of the two-phase data mining of spatial association rules. Our observation will be based on 4, 8, 16, 32, 64 and 128 Histogram Generators. For a certain dimension such as 4 HGs, four color values of HG_0, HG_1, HG_2 and HG_3 can be defined and their inter-color values are [0, 64], [65, 128], [129, 192] and [193, 255], respectively.

Figure 6 is a time comparison chart made for ten different images in different variances, under the circumstance that color count is 1000, the minimum support count is 40 and the minimum confidence is 0.001. Under the conditions that the variance is below 23.79 and the pre-determined Histogram Generator by the two-phase data mining is below 128, the time required for the data mining is obviously less compared to the Apriori. Under the conditions that the variances are between 97.51~ 828.681 and the pre-determined Histogram Generator by the two-phase data mining is below 16, the time required for the data mining is obviously less compared to the Apriori. With the increasing degrees of variance, our method may not be as effective. Therefore in observation of image variance, the two-phase data mining of spatial association rules is more effective compared to the Apriori, providing that the image variances are not as significant.

In addition to the above, our method also focuses on the effect from images of different pixels and color counts upon Histogram Generator counts. During the experiment, the defined minimum support count is 40 and the minimum confidence is 0.001. Figure 7 shows five images of different color counts and variances between 0.24~0.28. Figure 8 shows five images of different color counts and variances between 475.2~483.8. Figure 9 shows five images of different color counts and variances between 733~745. Vertical axis is the amount of time required. From the figures, the two-phase data mining of spatial association rules shows more promising effectiveness compared to the Apriori. By means of the results obtained from Figure 7, 8 and 9, Figure 10 allows the users to mine the spatial association rules according to the effective Histogram Generator index. This is to say, the users can pick on the Histogram Generator counts depending on images of different color counts. Say, when the color count in an image is 400, the users can pick on the Histogram Generator counts that are below 64 to establish more effective spatial association rules.

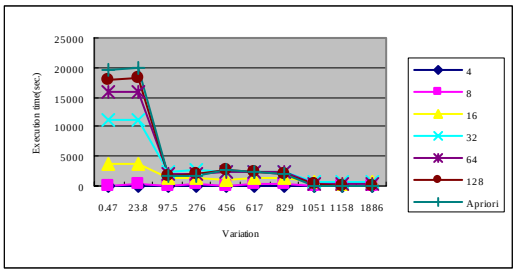


Figure 6. Relationship between different variance and time

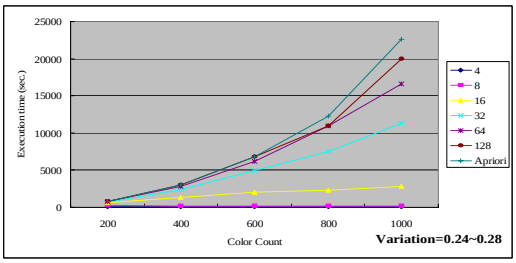


Figure 7. Execution time comparison between different HGs for some color count (variation=0.24~0.28)

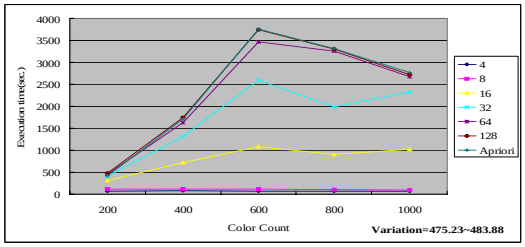


Figure 8. Execution time comparison between different HGs for some color count (variation=475.23~483.88)

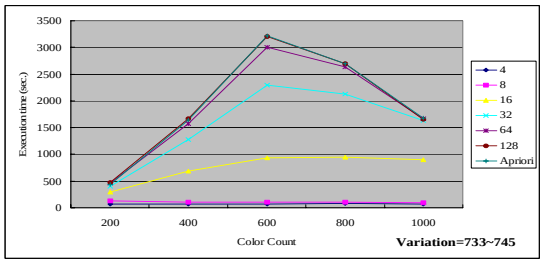


Figure 9. Execution time comparison between different HGs for some color count (variation=733~745)

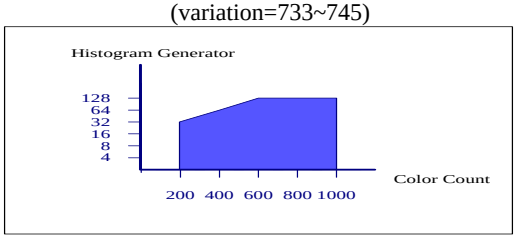


Figure 10. Relationship between Color and Histogram Generator

5-2. Experimental results of adaptive two-phase data mining of spatial association rules

Experimental analysis is made in section 4 on the adaptive two-phase data mining of spatial association rules. The experiment is mainly divided into two parts: First part is the time-related ratio between the adaptive two-phase data mining which incorporates the Histogram Generator as well as blocking and the Apriori (HG w.r.t. Partition)/Apriori. Second part is the time-related ratio between the adaptive two-phase data mining, the two-phase data mining of spatial association rules and the Apriori.

Figure 11 shows the time-related ratio between the adaptive two-phase data mining and the Apriori on five 50,000 pixels images of different color counts. The vertical axis is the time ratio (HG w.r.t. Partition)/Apriori. The time ratio less than 1 indicates that our method is effective. From Figure 11, the adaptive two-phase data mining under different color counts shows the time ratio being significantly less than 1; this is to day, the adaptive two-phase data mining is more effective than the Apriori.

Figure 12 shows the time-related ratio between the adaptive two-phase data mining which incorporates the Histogram Generator count that is 128 as well as blocking and the Apriori on five 50,000 pixels images under different color counts. Figure 12 also shows the time ratio between the two-phase data mining which incorporates the Histogram Generator that is 128 but excludes the blocking and the Apriori. When the time ratio is less than 1, it implies that the adaptive two-phase data mining is effective. From the Figure, the adaptive two-phase data mining yields the time ratio that is significantly less than 1; in other words, the adaptive two-phase data mining can improve the two-phase data mining of spatial association rules on its effectiveness.

Figure 13 is the time-related ratio between the adaptive two-phase data mining which incorporates the Histogram Generator as well as blocking and the Apriori on five 50,000 pixels images under different variances. The vertical axis is the time ratio (HG w.r.t. Partition)/Apriori. When the time ratio is less than 1, it implies that the adaptive two-phase data mining is effective. From the Figure, the adaptive two-phase data mining under different variances yields the time ratio that is significantly less than 1; in other words, the adaptive two-phase data mining is more effective than the Apriori.

Figure 14 is the time-related ratio between the adaptive two-phase data mining which incorporates the Histogram Generator count that is 128 as well as blocking and the Apriori. It also shows the time ratio between the two-phase data mining which incorporates the Histogram Generator count that is 128 but excludes the blocking and the Apriori. When the time ratio is less than 1, it implies that the adaptive two-phase data mining is effective. From the Figure, the adaptive two-phase data mining requires significantly less amount of time; in other words, image blocking in the first and coarse-grained association rules in the later are able to enhance the two-phase data mining in its effectiveness.

The biggest difference between the adaptive two-phase data mining and blocking [19] lies in that the method[19] must scan the database twice to obtain the final minimum support count in the global frequent itemsets and the adaptive two-phase data mining stores part of the data into the global non-frequent itemsets, so that there will be no need to go back and scan the database for the calculation of final minimum support count in the global frequent itemsets. Instead, only the adding calculation in the global non-frequent itemsets is required.

Figure 15 is a comparison chart between the database storage and the storage for the adaptive two-phase data mining to record the local non-frequent itemsets. During the entire experiment, the data counts are 1000k, the minimum support counts are 2000, the minimum confidence is 0.001, the partitioned block counts are 10 and the image variances are 6335.79~6522.68. From five images of different color counts in Figure 15, storage for the adaptive two-phase data mining to record the global non-frequent itemsets is far less than database storage; in other words, the adaptive two-phase data mining requires less storage because it does not have to account for the time on another database scanning.

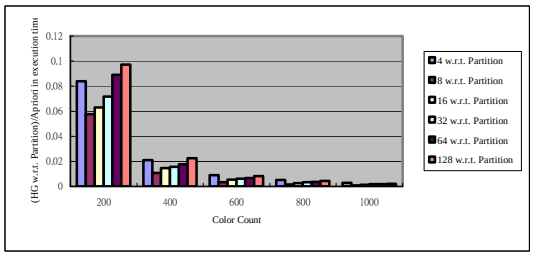


Figure 11. Execution time comparison between HG w.r.t. partition and Apriori for some color count

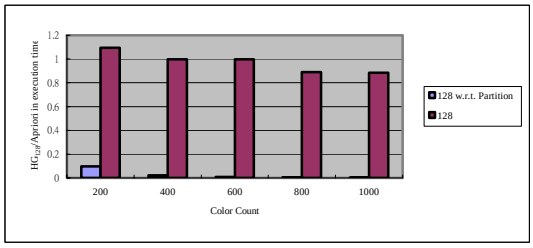


Figure12. Execution time comparison between 128 HGs w.r.t. Partition and 128 HGs for some color count

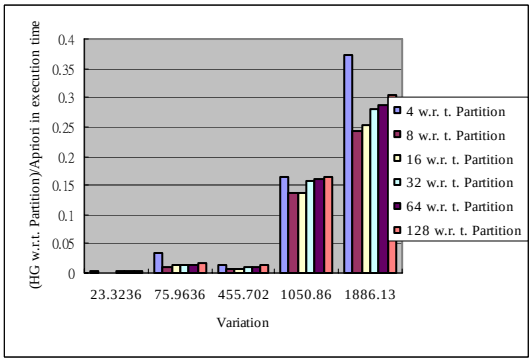


Figure 13. Execution time comparison between HG w.r.t. partition and Apriori for some variation

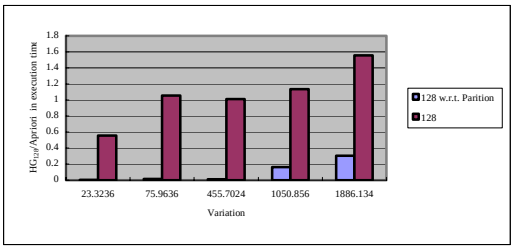


Figure 14. Execution time comparison between 128 HGs w.r.t. Partition and 128 HGs for some variation

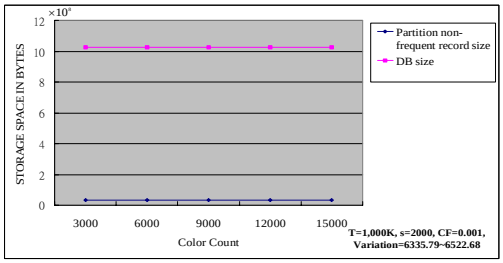


Figure 15. Storage comparison between non-frequent record size and DB size

6. CONCLUSIONS

Our proposed two-phase data mining method is essentially effective for the applications of data mining on remotely sensed images. The first step is to quickly find the coarse-grained association rules by Histogram Generator. In turn, find the fine-grained association rules from the previously mined association rules. This is especially efficient when a large amount of remotely sensed images is to be mined. Besides, the more concern will be placed on analysis of different types of images so that the users are still able to effective obtain the spatial association rules by our Histogram Generator index. To sum up, the mechanism of our method is more flexible and effective for data mining in images. Adaptive two-phase spatial association rules mining method conducts the idea of partition on an image for efficiently quantizing out non-frequent patterns and thus facilitate the following two phase process.

7. REFERENCES

[1] R. Agrawal, and R. Srikant (1994), "Fast Algorithms for Mining Association Rules in Large Database," Conference of Very Large Data Bases, Santiago, Chile, pp. 487-499.

[2] S. W. Changchien, and T. C. Lu (2001), "Knowledge Discovery from Object-Oriented Databases Using an Association Rules Mining Algorithm," Processing of the 5th International Conference on Knowledge-Based Intelligent Information Engineering System & Allied Technologies 6, 7, &8.

[3] E. Clementini, P. D. Felice, and K. Koperski (2000), "Mining Multiple-Level Spatial Association Rules for Objects with a Broad Boundary," Data and Knowledge Engineering, Vol. 34, pp. 251-270.

[4] A. Denton, W. Perrizo, Q. Ding, and Q. Ding (2002), "Efficient Hierarchical Clustering of Large Data Sets Using P-trees," Proceeding of 15th International Conference on Computer Applications in Industry and Engineering, San Diego, CA, pp. 138-141.

[5] Q. Ding, Q. Ding, and W. Perrizo (2002), "Decision Tree Classification of Spatial Data Streams Using Peano Count Trees," Proceeding of ACM Symposium on Applied Computing, Madrid, Spain, pp. 413-417.

[6] Q. Ding, Q. Ding, and W. Perrizo (2002), "Association Rule Mining on Remotely Sensed Images Using P-trees," Proceedings of PAKDD 2002, pp. 66-79.

[7] M. Ester, A. Frommelt, H. P. Kriegel, and J. Sander (1998), "Algorithms for Characterization and Trend Detection in Spatial Databases," Proceeding of 4th International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, pp. 44-50.

[8] M. Ester, S. Gundlach, H. P. Kriegel, and J. Sander (2000), "Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support," Data Mining and Knowledge Discovery, Vol. 4, pp. 193-216.

[9] M. Ester, and H. P. Kriegel (1997)**, "**Spatial Data Mining: A Database Approach," Processing 5th Int. Symposium on Large Spatial Databases, Berlin, pp. 47-66.

[10] M. Ester, H. P. Kriegel, and J. Sander (1999), "Knowledge Discovery in Spatial Databases," Conf. of 23rd German on Artificial Intelligence, Bonn, Germany, pp. 61-74.

[11] R. H. Guting (1994), "An Introduction to Spatial Database Systems," Conference of Very Large Data Base, pp. 357-400.

[12] J. Han, and Y. Fu (1995), "Discovery of Multiple-Level Association Rules from Large Databases," Processing 21th International Conference Very Large Data Bases, pp. 420-431.

[13] D. Holt, and M. Chung (2002), "Mining Association Rules Using Inverted Hashing and Pruning," Information Processing Letters, Vol. 83, No. 4, pp. 211-220.

[14] K. Koperski, and J. Han (1995), "Discovery of Spatial Association Rules in Geographic Information Databases," Proceeding Fourth Advances in Spatial Databases Symp. Springer, Berlin, pp. 47-66.

[15] K. Koperski, J. Han, and N. Stefanovic (1998), "An Efficient Two-Step Method for Classification of Spatial Data," Proceeding of International Symposium on Spatial Data Handling, Vancouver, BC, Canada, pp. 45-54.

[16] W. Lu, J. Han, and B. C. Ooi (1993), "Discovery of General Knowledge in Large Spatial Databases," Proceeding of Far East Workshop on Geographic Information Systems, World Scientific, Singapore, pp. 275-289.

[17] J. S. Park, M. S. Chen, and P. S. Yu (1995), "An Effective Hash-Based Algorithm for Mining Association Rules," Proceeding of ACM-SIGMOD Conference Management of Data, San Jose, CA, pp. 175-186.

[18] W. Perrizo, Q. Ding, Q. Ding, and A. Roy (2001), "Deriving High Confidence Rules from Spatial Data using Peano Count Trees," Proceedings of International Conference on Web-Age Information Management, Springer-Verlag, Lecture Notes in Computer Science 2118, pp. 91-102.

[19] A. Savasere, E. Omiecinski, and S. Navathe (1995), "An Efficient Algorithm for Mining Association Rules in Large Databases," Processing 21st VLDB Conference, pp. 432-444.

[20] K. Sayood (1996), "Introduction to Data Compression," Morgan Kauffman Publishers, San Fransisco, CA.

[21] Remote Sensing Tutorial Available http://rst.gsfc.nasa.gov/Front/tofc.html

Received: Nov 2005. Accepted: Feb. 2006.